**Interactive content 2**

**Final Exam**

Ans1:-

React JS:-

1. React is a library built by Facebook.
2. 2. React gives you only one thing: It's also all about components and all about building user interfaces from components.
3. React typically uses JavaScript.
4. React.js follows no strict release schedule but we see new versions being released
5. It is little difficult to learn than vue.

Angular JS:-

1. Angular is a framework developed by Google.
2. Angular out of the box includes support for a lot of things.
3. Angular projects use TypeScript, which is a superset to JavaScript.
4. For Angular, a new major version is released around every six months
5. It is difficult to learn

Vue JS:-

1. Vue is a "standalone" project
2. It's not as "big" as Angular but it definitely includes more features than React does.
3. Vue uses regular JavaScript and it typically utilizes something which is called "Single File Components".
4. Vue also is under active development.
5. It is easier to learn

Ans3:- SCSS is a special type of file for SASS, a program written in Ruby that assembles CSS style sheets for a browser, and for information, SASS adds lots of additional functionality to CSS like variables, nesting and more which can make writing CSS easier and faster.

Ans4:-

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

<th scope="col">Heading</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">1</th>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

<td>Cell</td>

</tr>

</tbody>

</table>

</div>

Ans5:- The technical difference between a framework and library lies in a term called inversion of control. When you use a library, you are in charge of the application flow. You choose when and where to call the library. When you use a framework, the framework is in charge of the flow. It provides you with a few places to plug in your code, but it calls the code you plugged in as needed.

Ans6:-

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio tag.

</audio>

Ans7:-

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

Ans8:- The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Ans9:-

1. **animation-name:** declares the name of the @keyframes at-rule to manipulate.
2. **animation-duration:** the length of time it takes for an animation to complete one cycle.
3. **animation-timing-function**: establishes preset acceleration curves such as ease or linear.
4. **animation-delay:** the time between the element being loaded and the start of the animation sequence (cool examples).
5. **animation-direction:** sets the direction of the animation after the cycle. Its default resets on each cycle.
6. **animation-iteration-count**: the number of times the animation should be performed.